Locally informed Gravitational Search Algorithm

Citation for published version:
Sun, G, Zhang, A, Wang, Z, Yao, Y, Ma, J & Couples, GD 2016, 'Locally informed Gravitational Search Algorithm', *Knowledge-Based Systems*, vol. 104, pp. 134–144. https://doi.org/10.1016/j.knosys.2016.04.017

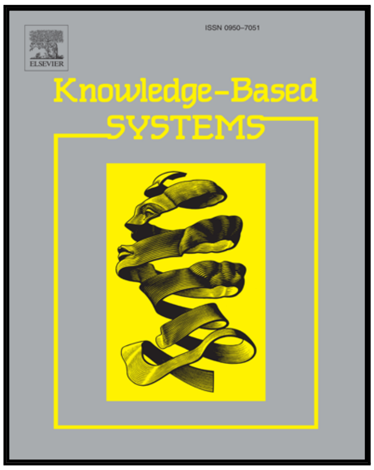
Digital Object Identifier (DOI):
10.1016/j.knosys.2016.04.017

Link:
Link to publication record in Heriot-Watt Research Portal

Document Version:
Peer reviewed version

Published In:
Knowledge-Based Systems

Accepted Manuscript



Locally informed Gravitational Search Algorithm

Genyun Sun, Aizhu Zhang, Zhenjie Wang, Yanjuan Yao,
Jinsheng Ma, Gary Douglas Couples

PII: S0950-7051(16)30059-4
DOI: 10.1016/j.knosys.2016.04.017
Reference: KNOSYS 3487

To appear in: *Knowledge-Based Systems*

Received date: 27 May 2015
Revised date: 29 March 2016
Accepted date: 17 April 2016

Please cite this article as: Genyun Sun, Aizhu Zhang, Zhenjie Wang, Yanjuan Yao, Jinsheng Ma, Gary Douglas Couples, Locally informed Gravitational Search Algorithm, *Knowledge-Based Systems* (2016), doi: 10.1016/j.knosys.2016.04.017

Locally informed Gravitational Search Algorithm

Genyun Sun[a,*], Aizhu Zhang[a], Zhenjie Wang[a], Yanjuan Yao[b], Jinsheng Ma[c], Gary Douglas Couples[c]

[a]*School of Geosciences, China University of Petroleum (East China), Qingdao Shandong, 266580, China*
[b]*Satellite Environment Center (SEC), Ministry of Environmental Protection (MEP) of China, Beijing, 100094, China*
[c]*Institute of Petroleum Engineering (IPE), Heriot-watt University, Edinburgh Scotland, EH14 2AD, United Kingdom*

Abstract

Gravitational search algorithm (GSA) has been successfully applied to many scientific and engineering applications in the past few years. In the original GSA and most of its variants, every agent learns from all the agents stored in the same elite group, namely K_{best}. This type of learning strategy is in nature a fully-informed learning strategy, in which every agent has exactly the same global neighborhood topology structure. Obviously, the learning strategy overlooks the impact of environmental heterogeneity on individual behavior, which easily resulting in premature convergence and high runtime consuming. To tackle these problems, we take individual heterogeneity into account and propose a locally informed GSA (LIGSA) in this paper. To be specific, in LIGSA, each agent learns from its unique neighborhood formed by k local neighbors and the historically global best agent rather than from just the single K_{best} elite group. Learning from the k local neighbors promotes LIGSA fully and quickly explores the search space as well as effectively prevents premature convergence while the guidance of global best agent can accelerate the convergence speed of LIGSA. The proposed LIGSA has been extensively evaluated on 30 CEC2014 benchmark functions with different dimensions. Experimental results reveal that LIGSA remarkably outperforms the compared algorithms in solution quality and convergence speed in general.

Keywords: Gravitational Search Algorithm (GSA), Environmental Heterogeneity, k-neighborhood Local Search, Locally Informed Learning

1. Introduction

Evolutionary algorithms and population-based optimization algorithms have been widely used for solving various optimization problems in the past decades [4, 5, 7–9, 11, 14, 20, 34]. Gravitational search algorithm (GSA) is one of the latest population-based optimization algorithms, which is inspired from the Newton's law of gravity and motion[29]. In GSA, the performance of an agent is measured by its mass. The heavy masses correspond to good solutions. As Newtonian gravity states that "Every agent in the universe attracts every other agent with a force that is directly proportional to the product of their masses and inversely proportional to the square of the distance between them", the relevant force will cause a global movement of each agent towards those agents with heavier masses [25, 41]. Hence, an agent can search for the global optimum iteratively by learning from all of the rest agents. In essential, it is a type of fully-informed learning strategy in nature, which makes GSA has an outstanding property: diverse search directions.

Although the fully-informed learning strategy is simple in theory and easy to use, it easily causes two problems: 1) suffering from high runtime consuming [2] and 2) performing a poor tradeoff between exploration and exploitation [29]. On one hand, for a population with N agents, to obtain the force of an agent exerted by the rest agents, N-1 times distance should be calculated. Consequently, performing one iteration in the population, $N(N-1)$ times distances between agents need to be computed, which results in high runtime consuming [2]. On the other hand, the fully-informed learning strategy makes each agent learns from the rest agents in all the time, which means every agent exactly has the same global neighborhood topology structure [40]. This type of global structure overly emphasis on exploitation and offends against the basic rules of population-based optimization algorithm: to achieve well balance, exploration must fade out and exploitation must fade in by the lapse of time [29, 31].

*Corresponding author
Email address: genyunsun@163.com (Genyun Sun)

To obtain compromise between exploration and exploitation, a K_{best} model is employed in original GSA. The K_{best} model stores those superior agents after fitness sorting in each iteration. The size of K_{best} is a function of time, which is set to N at the beginning and linearly decreases with time down to one. In such a way, each agent is guided by the rest agents at the beginning while only by one agent at the end [29].

Although the K_{best} model plays a certain effect, some problems still remain. On one hand, the overall computational time of GSA is still high as the size of K_{best} decreasing slowly. On the other hand, in the later stages, each agent can only learn from few elite agents, which easily causes quick loss of search diversity and false convergence. In this case, once the prematurity occurs, the population will trap into local optima because there are no remedies. Moreover, this model weakens the role of the global best agent due to the fact that all the elite agents have the equal status in K_{best}. Especially, the historically global best agent is discarded once the population is updated. GSA therefore ignores the importance of the global best agent in guaranteeing the convergence speed and accuracy [27]. The biggest problem lies in the topology structure of K_{best} model that is still a global neighborhood topology structure, in which every agent learns form the same group of elite. Due to the single topology structure, GSA overlooks the influence of environmental heterogeneity on individual behavior.

In the past few years, many researches have focused on improving GSA. One active research trend is to introduce some new operators into the original GSA. In [32], a disruption operator was employed to further explore and exploit the search space. Then, Shaw et al. [33] used opposition-based learning to perform population initialization and generation jumping, and improved the exploitation ability of GSA in the last iterations. In [10], the Black Hole theory was utilized to prevent premature convergence and to improve the exploration and exploitation abilities of GSA. Another active research trend is to combine some state-of-art heuristic optimization algorithms with GSA. For example, Li et al. [22] integrated Differential Evolution (DE) into GSA to overcome the premature convergence existing in unconstrained optimization. Sun et al. [35] presented a hybrid GA and GSA (GAGSA) to overcome the premature convergence problem. In addition, the memory of particle swarm optimization (PSO) has been introduced to GSA for constructing some more promising variants of GSA. In the PSOGSA [25, 26] and GGSA [27], social thinking was introduced to GSA to accelerate convergence speed in the last iterations. In gravitational particle swarm [37] and modified GSA [13], the movement of each agent is determined by velocity of PSO and acceleration of GSA. In improved GSA [16, 17], both the chaotic perturbation operator and memory of the position of each agent were utilized. The chaotic operator can enhance its global convergence to escape from local optima, and the memory strategy provides a faster convergence and shares individual best fitness history to improve the search ability.

Essentially, most of the GSA variants mentioned above are presented to enhance the search performance of GSA by designing new learning strategies or promoting the population diversity. However, most of them treat every agent equally, i.e., every agent learns from the same elite group stored in the K_{best}. In other words, the sight range of each agent is exactly the same, which disregarding the local environment of agents and easily resulting in premature convergence and high runtime consuming.

The aforementioned issues prompt us to explore the effect of environmental heterogeneity on individual behavior and proposed a GSA variant called locally informed GSA (LIGSA). The novelties of LIGSA are in two areas as follows.

(1) A locally informed learning strategy is proposed. The environmental heterogeneity is taken into account by constructing unique local neighborhood for each agent. Learning from the k local neighbors promotes LIGSA fully explore the regions around each agent with low computational complexity as well as effectively prevent premature convergence.

(2) Historical experience of the population is introduced to GSA. Each agent can learn from the historically global best agent directly. This makes the historically global best agent play a remarkable role for guiding the convergence process. Thereby the convergence speed of LIGSA is accelerated.

The remainder of this paper is organized as follows. Section 2 briefly describes the framework of GSA as well as discusses the fully-informed learning mechanism of GSA. In Section 3, a detail introduction of the proposed LIGSA is given. The comparison experimental results and discussion are presented in Section 4. Finally, a conclusion is given in Section 5.

2. Overview of GSA

2.1. GSA Framework

In GSA, every agent $\boldsymbol{x}_i = [x_{i1}, x_{i2}, ..., x_{iD}]$ $(i = 1, 2, ..., N)$ attracts each other by gravitational force in a D-dimensional search space according to the law of gravity [29]. The corresponding velocity of agent i is $\boldsymbol{v}_i = [v_{i1}, v_{i2}, ..., v_{iD}]$. Due to the force between two agents is directly proportional to their masses and inversely proportional to their distance, all the agents move towards those agents that have heavier masses [29, 30]. The mass of each agent in generation t, denoted by M_i^t, is simply calculated by Eqs. (1) and (2) as follows:

$$mass_i^t = \frac{fit_i^t - worst^t}{best^t - worst^t}, \tag{1}$$

$$M_i^t = \frac{mass_i^t}{\sum_{j=1}^{N} mass_j^t}, \tag{2}$$

where fit_i^t represents the fitness value of the agent i in generation t. For a minimization problem, $worst^t$ and $best^t$ are defined in Eqs. (3) and (4) as follows:

$$best^t = \min_{j \in [j=1,2,...,N]} fit_j^t, \tag{3}$$

$$worst^t = \max_{j \in [j=1,2,...,N]} fit_j^t. \tag{4}$$

In an optimization problem, the force acting on the agent i from agent j at a specific time t is shown in Eq. (5) as follows:

$$F_{ij}^d(t) = G\,(t)\,\frac{M_i^t\,(t)\,M_j^t\,(t)}{R_{ij}^t\,(t) + \varepsilon}\left(x_{jd}^t\,(t) - x_{id}^t\,(t)\right), \tag{5}$$

where $G(t)$ is the gravitational constant in generation t, M_i^t and M_j^t are the gravitational mass of the agents i and j, x_{jd}^t is the position of the agent j and x_{id}^t represent the position of the agent i in the d-th dimension, respectively. R_{ij}^t is the distance between the agents i and j, and ε is a small constant bigger than 0.

To give a stochastic characteristic to GSA, the total force that acts on the agent i in the d-th dimension is set to be a randomly weighted sum of d-th components of the forces exerted from other agents as shown in Eq. (6) as follows:

$$F_i^d(t) = \sum_{j=1, j\neq i}^{N} rand_j F_{ij}^d(t), \tag{6}$$

where $rand_j$ is a uniform random variable in the interval [0, 1].

Hence, by the law of motion, the acceleration of the agent i in generation t, and in the d-th dimension, a_{id}^t, is given in Eq. (7) as follows:

$$a_{id}^t = \frac{F_{id}^t}{M_i^t}. \tag{7}$$

The gravitational constant, G, is initialized to G_0 at the beginning and decreases with time to control the search accuracy. It is defined in Eq. (8) as follows:

$$G(t) = G_0 \cdot e^{-\beta \frac{t}{T_{\max}}}, \tag{8}$$

where β is the coefficient of decrease and $T_{\max}$ is the maximum number of iterations. In the original GSA, G_0 is set to 100 and β is set to 20. This setting is adopted by all the GSA variants of this paper.

In generation t, the velocity and the position of the agent i are updated according to Eqs. (9) and (10) as follows:

$$v_{id}^{t+1} = rand_i \times v_{id}^t + a_{id}^t, \tag{9}$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}, \tag{10}$$

where v_{id}^t is the velocity of the agent i in the d-th dimension, $rand_i$ is a uniform random variable in the interval [0, 1].

2.2. Fully-informed Learning Mechanism in GSA

Following the Newton's law of gravity and motion, each agent in GSA is attracted by every each other as illustrated in Eqs. (5)-(10). Due to force between two agents is directly proportional to their masses and inversely proportional to the square of the distance between them, the relevant force on a target agent will be guided move towards the heaviest mass [29] as shown in Fig. 1. In nature, this kind of learning strategy that all agents involved are often referred to as fully-informed learning strategy. This learning strategy equips GSA with diverse search directions.

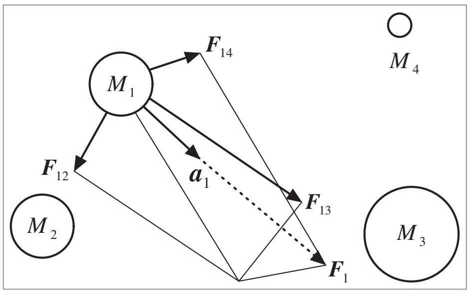


Figure 1: Schematic diagram of agent's movement in GSA.

However, the global neighborhood topology structure of the fully-informed learning strategy causes quick loss of the population diversity and high runtime consuming [2, 18]. As a result, the exploration ability of GSA decreases rapidly by the lapse of iterations and hence GSA is easily falling into local optima. One way to perform a good balance between exploration and exploitation is to reduce the number of agents with lapse of time in Eq. (6). Therefore, a K_{best} model is introduced and Eq. (6) was modified to Eq. (11) as follows:

$$F_i^d(t) = \sum_{j \in K_{\text{best}}, j \neq i}^{N} rand_j F_{ij}^d(t), \tag{11}$$

where K_{best} is the set of first K agents with better fitness value and bigger mass and the size of K_{best} is gradually decreased with the lapse of iterations.

As the size of K_{best} decreased slowly, the overall computational time of GSA is still high. Moreover, in the final iterations, the algorithm convergence to the current optimum too fast and performs poor local search ability. In addition, the guidance of the current global best agent is unremarkable because the current agent is attracted by all its neighbors. If one agent is significantly nearer to the local optimum agent than global optimum agent as shown in Fig. 2, the force exerted by the global optimum agent will be extremely small while the force exerted by the local optimum is considerably large. In this case, the search direction of M_1 will tend to the local optimum M_2, and premature convergence is easy to happen.

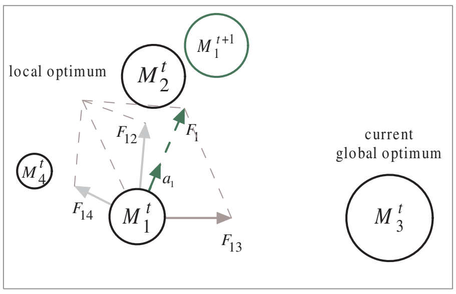


Figure 2: Schematic diagram of falling into a local optimum.

Most importantly, the K_{best} model is still a type of global topology based fully-informed method. In this model, all the agents exactly have the same global neighborhood topology structure and learn from the same group of neighbors (elite) stored in K_{best}. Therefore, the impact of environmental heterogeneity on individual behavior is overlooked and thus resulting in premature convergence and high runtime consuming. To improve the efficiency of GSA, we take the environmental heterogeneity of each individual into consideration and propose a new local search based GSA variant, namely LIGSA.

3. Proposed LIGSA

This section describes the locally informed gravitational search algorithm (LIGSA) which is designed based on the environmental heterogeneity of agents. First of all, the k-neighborhood topology and social thinking of population are integrated to build a locally informed learning strategy as described in Section 3.1. Then, the technical process of LIGSA is given point-by-point in Section 3.2. The LIGSA first creates the population (candidate solutions) with in preassigned search space using a random initialization strategy. Then the fitness value of each agent is evaluated. Based on the fitness values, the global best agent can be determined. Afterward, the mass of each agent and gravitational constant is updated according to the current iteration and fitness values. And then, we utilized the proposed locally informed learning strategy to update the velocity and position of each agent. Subsequently, three schemes are employed to eliminate duplicate agents and guarantee the boundary constraints. Finally, the best solution will be outputted when the optimization phrase encounter the termination conditions.

3.1. Locally Informed Learning Strategy

In this section, the locally informed learning strategy is introduced. In the new learning strategy, each agent is guided by 1) the resultant force exerted by all the agents in its local neighborhood and 2) the historically global best agent, denoted by $\boldsymbol{gbest} = [g_1, g_2, ..., g_D]$. The velocity updating rule is thus very different from the Eq. (9) of GSA, as shown in Eq. (12):

$$v_{id}^{t+1} = rand_i \times v_{id}^t + \sum_{j \in K_{\text{local}}^i} rand_j G(t) \frac{M_j^t}{R_{ij}^t + \varepsilon}(x_{jd}^t - x_{id}^t) + (g_d^t - x_{id}^t), \tag{12}$$

where K_{local}^i is a new presented k-neighborhood (wheel topology), called locally informed k-neighborhood as shown in Fig. 3. In contrast with traditional k-neighborhood topology in which only the best agent is chosen to perform guidance [24], in K_{local}^i, all the agents are associate with the current agent i. Thus, the agent i can fully learn from its k neighbors through the gravitational force, which preserving the diverse search directions.

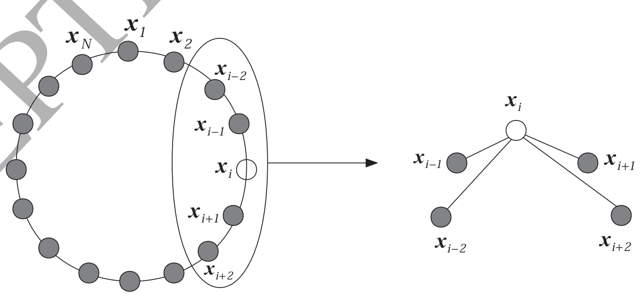


Figure 3: The locally informed k-neighborhood (k=4).

Obviously, in LIGSA, each agent has a unique neighborhood K_{local}^i and thus the environmental heterogeneity of each individual is considered. By contrast, in original GSA, all the agents learn from the same elite group, K_{best}, which overlook the impact of environmental heterogeneity on individual behavior. Moreover, the K_{best} model is a type of global neighborhood topology while the K_{local}^i is a local neighborhood topology. The local neighborhood topology has been proven more skilled in complicated problems than the global topology [6]. Benefit from these properties, LIGSA can perform preeminent local search ability as well as effectively reduce the computational complexity.

In the locally informed k-neighborhood, the size of neighborhood may influence the exploration and exploitation abilities of LIGSA. The existing researches also indicate that a smaller neighborhood may be more suitable for complex problems while a larger may perform better on simple problems [38]. However, too small neighborhood will leads to poor diversity of search direction. This may weaken the exploration ability of GSA. On the other hand, if the number of neighbors is too large, the computational complexity will be high. An appropriate size of neighborhood is about 15 percent of the population size as suggested in many applications [12]. Hence, to make the search ability of GSA more flexible, especially for solving complex problems, as well as to reduce the computational complexity, $k = 2 * \lfloor 15\% * N/2 \rfloor$ is chosen in this paper. It is worth noting that the k-neighborhood, K^i_{local} of the agent i is defined as $K^i_{\text{local}} = \{\boldsymbol{x}_{i-k/2}, \boldsymbol{x}_{i-k/2+1}, ..., \boldsymbol{x}_{i-1}, \boldsymbol{x}_{i+1}, ..., \boldsymbol{x}_{i+k/2-1}, \boldsymbol{x}_{i+k/2}\}$ according to its index. Fig. 3 is a 4-neighborhood example in which $K^i_{\text{local}} = \{\boldsymbol{x}_{i-2}, \boldsymbol{x}_{i-1}, \boldsymbol{x}_{i+1}, \boldsymbol{x}_{i+2}\}$.

The third component $(g^t_d - x^t_{id})$ of Eq. (12) shows the role of the historically global best agent. With this mechanism, the previous search information of the population can be utilized. In other words, the role of the historically global best agent is remarkable. Different from the local neighborhood topology of the second component, the guidance of ***gbest*** is exhibited as a global neighborhood topology. According to [2, 3], the global neighborhood topology based algorithms show a better performance than local search based algorithms in unimodal problems while the local search based algorithms provide good results in multimodal. Hence, both local and global topologies are incorporated in LIGSA to achieve preeminent optimization. Moreover, the historically global best agent guides the agent i based on their position difference, in which the step length is changed automatically associated with the convergence stages. This will effectively accelerate the convergence speed of LIGSA.

In addition, population-based algorithms require the emphasis of exploration in the first iterations and exploitation in the final iterations. Since there is no clear border between the exploration and exploitation phases, the adaptive method is the best option for allowing a gradual transition between these two phases [27]. Therefore, two time-varying acceleration coefficients are introduced to Eq. (12) in this paper. The velocity and position update equations in LIGSA are thus designed by Eqs. (13) and (14) as follows:

$$v^{t+1}_{id} = rand_i \times v^t_{id} + c_1 \cdot \sum_{j \in K^i_{\text{local}}} rand_j G(t) \frac{M^t_j}{R^t_{ij} + \varepsilon} (x^t_{jd} - x^t_{id}) + c_2 \cdot (g^t_d - x^t_{id}), \tag{13}$$

$$x^{t+1}_{id} = x^t_{id} + v^{t+1}_{id}, \tag{14}$$

where c_1 and c_2 are adaptively adjusted according to the iteration. In order to fully explore the search space and accelerate convergence in the last iterations, c_1 and c_2 are defined in Eqs. (15) and (16) as follows:

$$c_1 = 1 - t^3/T^3_{\max}, \tag{15}$$

$$c_2 = t^3/T^3_{\max}, \tag{16}$$

where t is the current iteration time and $T_{\max}$ is the maximum iterations. A diagrammatic sketch of the time-varying acceleration coefficients scheme is shown in Fig. 4.

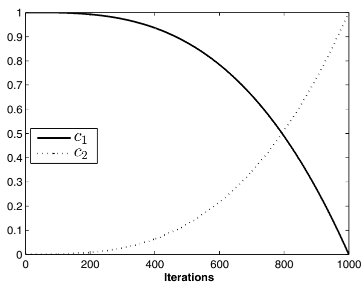


Figure 4: Curves of the two time-varying acceleration coefficients.

Apparently, these acceleration coefficients are changed automatically with the convergence stage. In the early stages, the agents explore the search space more globally for c_1 is greater than c_2. While in the later stages, agents tend to convergence toward the global best agent quickly for c_2 is greater than c_1.

3.2. Technical Process of LIGSA

The detailed steps of LIGSA are described in the following subsections.

3.2.1. Population initialization

Population initialization, including position and velocity initialization of each agent, is the first and the primary task in any evolutionary algorithm [36]. In LIGSA, to spread the agents as extensive as possible in the search space, the initial positions are set randomly in the range of the search space as shown in Eq. (17):

$$x_{id} = rand * (ub_d - lb_d), d \in [1, 2, ..., D], i \in [1, 2, ..., N], \tag{17}$$

where N is the size of the population, D is the dimension of the search space, ub_d is the upper bound and lb_d is the lower bound of the search space in the d-th dimension. The corresponding initial velocity of each agent $\boldsymbol{v}_i = [v_{i1}, v_{i2}, ..., v_{iD}]$ is set to zero in this study.

3.2.2. Evaluate each agent in the population

For a single objective optimization problem, the fitness value of each agent is calculated on the basis of its function equation. Simultaneously, the function values of all the constraints for each agent are calculated if the user aimed to solving constraint problems.

3.2.3. Update global best agent

For a minimum problem, the global best agent $\boldsymbol{gbest}$ is updated according to its fitness values. That is, if fitness value of the best agent in the current population is not bigger than $\boldsymbol{gbest}$, then its position is replaced; otherwise, the $\boldsymbol{gbest}$ in memory is kept. Apparently, this kind of elite lets LIGSA has the global thinking characteristic. Thus, the previous experience of the population is kept and utilized to guide the search of population agents.

3.2.4. Update agents' velocities and positions

In this phase, we need to calculate the gravitational constant and masses of agents first. The update of gravitational constant follows the Eq. (8) and G_0 is set to 100 in the beginning. The calculation of agent mass follows the methods used in GSA as shown in Eqs. (1)-(4). Moreover, according to the description in Section 3.1, the velocity and position update in LIGSA is carried on by Eqs. (13) and (14).

3.2.5. Duplicates removal and boundary constraint operators

After the population update is completed, the positions of some agents may be exactly the same as some other agents. Consequently, the diversity of the population will decrease. This yields some negative effects to the convergence accuracy. To eliminate the duplicate agents, the randomly initialization method shown in Eq. (17) is used to update their positions. If the agents i and j are duplicates, either of them is selected, and then modify its position follows Eq. (17) while its velocity is kept.

In addition, for all of the population-based optimization algorithms, some agents may fly out of the search space and make the population misses the optimal solution in the search space [12, 15]. To restrict agents in the search space, we randomly relocate those fly-out agents within the search space like many researches [12, 28].

The velocity is an important factor causing agents to fly out of the search spacey. In LIGSA, we introduce an operator to control the velocity bounds of the agents by Eq. (18) [19, 28] as follows:

$$v\text{max}_d = (ub_d - lb_d)/N_{\text{limit}}, v\text{min}_d = v\text{max}_d, \tag{18}$$

where N_{limit} can be any positive integer.

In LIGSA, after the position update if the velocity of an agent exceeds the boundary constraints, its velocity must be set to the corresponding critical values. If we take a high value of N_{limit}, the velocity bounds will be small. Consequently, for any high velocity the agents will be randomly relocated and lose its normal momentum and path [28]. So we have taken a small value of N_{limit}, here N_{limit}=2 to have safe limits.

3.2.6. Termination conditions

The optimization process stops when the termination conditions are met. The conditions usually set by user according to demands. Normally, maximum number of iterations or maximum number of fitness evaluations ($FES_{\max}$) can be used as the termination criterion. In this study, $FES_{\max}$ is used as the termination condition for the proposed LIGSA and the other 6 comparison algorithms. If the algorithm reaches the termination condition, the optimization process stops, and the final best solution is obtained. Otherwise, the optimization process goes to agent evaluation in Section 3.2.2 and executes iteration along with Section 3.2.3 to Section 3.2.5.

The overall flowchart of the proposed LIGSA is shown in Fig. 5.

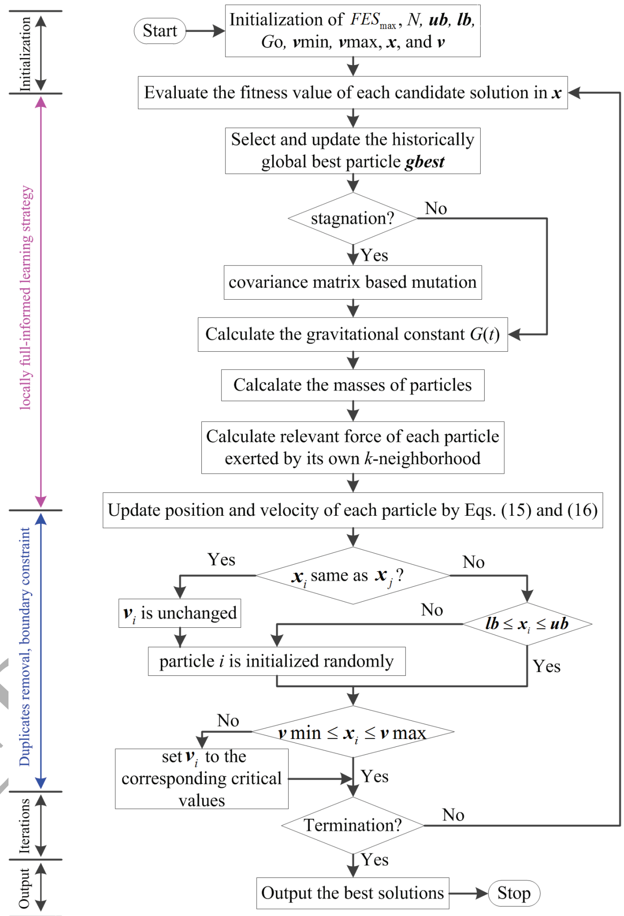


Figure 5: Flowchart of LIGSA.

4. Experimental and Discussion

4.1. Experimental Setup and Results

For the evaluation of LIGSA, a comprehensive experimental evaluation and comparison with the original GSA [29], four competitive variants of GSA (GGSA [27], PSOGSA [25], GAGSA [35], MGSA [13]), and a local PSO (LPSO) is provided based on 30 benchmark functions of the CEC 2014. The detailed description of these functions can be found in [23]. Among the compared algorithms, GAGSA is a recently published GSA variant, GGSA and PSOGSA integrated the global best agent in updating velocity while MGSA used both the personal best and global best in the iteration process.

During the experiments, to perform fair comparisons, for all of the 7 algorithms, the N and $FES_{\max}$ were set to 60 and 60 000, respectively. Meanwhile, the G_0, β, and k in all the 5 compared GSA variants were set to 100, 20 and [N to 1], respectively. For the four GSA variants, all the other parameters have been set as suggested by authors. For LIGSA, the acceleration coefficients were set to $c_1 = 1 - t^2/T^3_{\max}$ and $c_2 = t^3/T^3_{\max}$ the number of neighbors was set to $2 * \lfloor 15\% * N/2 \rfloor$ as discussed in Section 3.1. For LPSO, the size of local neighborhood was set to 4, ω was decreased linearly from 0.9 to 0.4, and c_1 & c_2 were defined as FL-GSA. The parameter settings of the 7 algorithms are provided in Table 1.

Table 1: Parameter settings in this paper.

	LIGSA	GSA	GAGSA	PSOGSA	GGSA	MGSA	LPSO
$FES_{\max}$	60 000	60 000	60 000	60 000	60 000	60 000	60 000
N	60	60	60	60	60	60	60
G_0	100	100	100	100	100	100	100
β	20	20	20	20	20	20	20
ω	–	–	–	–	–	–	[0.9 to 0.4]
c_1	$1 - t^3/T^3_{\max}$	–	–	0.5	$2 - 2t^3/T^3_{\max}$	0.5	$1 - t^3/T^3_{\max}$
c_2	$t^3/T^3_{\max}$	–	–	1.5	$2t^3/T^3_{\max}$	0.5	$t^3/T^3_{\max}$
k	$2 * \lfloor 15\% * N/2 \rfloor$	[$N$to 2]	[N to 2]	[N to 2]	[N to 2]	[N to 2]	$2 * \lfloor 15\% * N/2 \rfloor$

To fully evaluate the LIGSA, comparison between LIGSA and the other 6 algorithms is performed based on 30 and 50 dimensional versions of the CEC2014 benchmark functions (D=30 and D=50). The average (Mean), standard deviation (Dev), and best (Best) of the optimization error (best-optimum) of 30 independent runs of each algorithm were presented in Tables 2-6. For each performance metric, the best obtained results were shown in boldface.

In addition, to further compare and statistically analyze the obtained optimization results, two-sample t-tests were also conducted in this section. Two-sample t-test is a hypothesis testing method for determining the statistical significance of the difference between two independent samples of an equal sample size [39]. In this paper, the significance level is α=0.05 and the free degree is 29. Therefore, if in any test a t-value that is smaller than or equal to critical value '-2.045' is produced, the alternative hypothesis is considers a significant difference of both approaches. Meanwhile, the corresponding t-value was depicted in boldface. For an easy observation, the summary of the t-test are reported in Table 7. In Table 7, "better" indicates LIGSA outperforms the compared algorithm significantly while "same" suggests that the superiority of LIGSA is not significantly. The word "worse" implies that LIGSA produced significantly worse results on the tested functions.

As illustrated in Table 2, for the *unimodal functions* (F1-F3), LIGSA outperformed all the 6 compared algorithms on all the three performance metrics when the dimension is high D=50. For low dimension (D=30) unimodal problems, although the average performance of LIGSA is in the second place and poor than GGSA on F2, it yielded the best optimization error. Moreover, the superiority of LIGSA is significantly on almost all the cases as stated by the t-test value in Table 7.

The next 13 function are *multimodal functions* (F4-F16). They are shifted and rotated functions with numbers of local optima, in which the global optimum is more difficult to locate. Form the results presented in Tables 2-4 we can conclude that although the mean and best errors of LIGSA are worse than GSA and GGSA on F5, LIGSA achieved superior results on most other functions. To be specific, for the low dimension (D=30), LIGSA produced the highest mean accuracy on 9 out of the 13 *multimodal functions*, including F4, F6, F8-F10, F12-F14, and F16. Similarly, for the high dimension (D=50), LIGSA also performed best optimization results on 9 out of the 13 *multimodal functions*, including F6-F9, F11-F12, F14-F16. Furthermore, in terms of t-test, LIGSA significantly outperformed the other comparison algorithms in all dimensions on F5-F6, F8-F9, F12, F14 and F16.

Table 2: Optimization errors of the CEC2014 benchmark functions (F1-F6) with D=30 & D=50.

Algorithm	function	F1		F2		F3		F4		F5		F6	
	D	30	50	30	50	30	50	30	50	30	50	30	50
LIGSA	Mean	**4.49E+05**	**4.66E+06**	1.15E+06	**1.72E+04**	**1.42E-01**	**7.09E+03**	**6.09E+01**	**9.82E+01**	2.08E+01	2.10E+01	**3.59E+00**	**1.33E+01**
	Best	**2.13E+05**	**3.10E+06**	**1.26E+02**	**8.53E+03**	**5.45E-02**	**2.82E+03**	**3.49E+00**	**9.36E+01**	2.07E+01	2.10E+01	**1.00E+00**	**1.05E+01**
	Dev	**1.97E+05**	**1.64E+06**	2.52E+06	**8.18E+03**	**1.36E-01**	**4.73E+03**	3.25E+01	**3.11E+00**	6.02E-02	4.14E-02	1.91E+00	2.74E+00
	t-test		–	–	–	–	–	–	–	–	–	–	–
GSA	Mean	1.13E+08	3.42E+08	9.81E+08	3.01E+10	7.57E+04	1.42E+05	2.89E+02	4.02E+03	**2.00E+01**	**2.00E+01**	2.75E+01	5.58E+01
	Best	8.94E+07	2.52E+08	6.65E+08	2.49E+10	7.17E+04	1.24E+05	2.57E+02	3.48E+03	**2.00E+01**	**2.00E+01**	2.36E+01	5.13E+01
	Dev	2.12E+07	8.42E+07	3.07E+08	3.82E+09	3.57E+03	1.41E+04	2.81E+01	4.57E+02	**1.45E-04**	1.73E-04	2.86E+00	3.09E+00
	t-test	**-11.939**	**-8.95914**	**-7.136**	**-17.6147**	**-47.438**	**-20.243**	**-11.856**	**-19.205**	29.366	53.981	**-15.565**	**-22.979**
GAGSA	Mean	1.78E+09	7.12E+09	8.07E+10	1.65E+11	8.51E+04	4.62E+05	1.60E+04	5.40E+04	2.11E+01	2.12E+01	4.52E+01	7.91E+01
	Best	1.41E+09	5.23E+09	6.48E+10	1.42E+11	8.39E+04	1.86E+05	1.43E+04	4.69E+04	2.10E+01	2.12E+01	4.39E+01	7.49E+01
	Dev	2.41E+08	1.69E+09	1.03E+10	1.31E+10	9.37E+02	3.29E+05	1.01E+03	4.81E+03	4.88E-02	3.40E-02	**1.06E+00**	**2.68E+00**
	t-test	**-16.571**	**-9.4228**	**-17.509**	**-28.102**	**-203.168**	**-3.090**	**-35.146**	**-25.032**	**-8.688**	**-9.427**	**-42.577**	**-38.345**
PSOGSA	Mean	2.16E+08	3.96E+08	1.41E+10	5.05E+10	1.03E+05	2.52E+05	9.44E+02	8.41E+03	2.01E+01	2.04E+01	2.28E+01	5.31E+01
	Best	5.11E+07	1.18E+08	1.45E+09	2.49E+10	3.96E+04	1.36E+05	2.46E+02	4.48E+03	**2.00E+01**	2.00E+01	1.96E+01	4.86E+01
	Dev	1.59E+08	2.23E+08	1.76E+10	1.81E+10	7.00E+04	8.50E+04	8.52E+02	5.18E+03	1.42E-01	2.32E-01	2.03E+00	4.30E+00
	t-test	**-3.0419**	**-3.93596**	-1.787	**-6.22694**	**-3.281**	**-6.429**	**-2.317**	**-3.583**	10.276	6.095	**-15.388**	**-17.431**
GGSA	Mean	5.09E+07	4.94E+07	**4.46E+03**	6.95E+08	6.39E+04	1.25E+05	1.34E+02	4.32E+02	**2.00E+01**	**2.00E+01**	1.45E+01	4.15E+01
	Best	4.28E+07	3.37E+07	1.90E+03	1.88E+08	5.24E+04	1.07E+05	9.04E+01	3.86E+02	**2.00E+01**	**2.00E+01**	1.23E+01	3.67E+01
	Dev	7.40E+06	1.07E+07	**2.38E+03**	7.59E+08	7.72E+03	1.24E+04	3.26E+01	4.23E+01	2.06E-04	**1.20E-04**	2.10E+00	3.19E+00
	t-test	**-15.246**	**-9.229**	1.016	**-2.04793**	**-18.490**	**-19.931**	**-3.536**	**-17.638**	29.371	53.981	**-8.611**	**-15.001**
MGSA	Mean	1.55E+07	2.45E+07	6.71E+03	2.42E+05	9.15E+04	1.54E+05	1.04E+02	3.18E+02	2.05E+01	2.10E+01	2.05E+01	4.00E+01
	Best	8.48E+06	9.80E+06	6.90E+02	1.10E+05	3.33E+04	9.93E+04	7.02E+01	2.28E+02	**2.00E+01**	2.09E+01	1.74E+01	3.49E+01
	Dev	8.17E+06	1.07E+07	6.15E+03	1.29E+05	4.56E+04	5.19E+04	**3.19E+01**	8.26E+01	4.39E-01	6.71E-02	2.27E+00	3.16E+00
	t-test	**-4.131**	**-4.104**	1.014	**-3.8775**	**-10.273**	**-6.297**	**-2.095**	**-5.938**	1.316	-0.173	**-12.748**	**-14.252**
LPSO	Mean	1.61E+08	6.85E+08	4.00E+10	9.23E+10	9.15E+04	2.36E+05	2.07E+03	9.63E+03	2.09E+01	2.10E+01	3.62E+01	6.62E+01
	Best	2.63E+07	2.68E+08	2.42E+10	7.14E+10	3.33E+04	1.51E+05	1.70E+03	3.83E+03	2.07E+01	2.09E+01	2.62E+01	5.62E+01
	Dev	1.36E+08	2.44E+08	1.41E+10	1.28E+10	4.56E+04	9.24E+04	4.46E+02	4.21E+03	9.01E-02	6.80E-02	5.76E+00	6.05E+00
	t-test	**-2.642**	**-6.240**	**-6.355**	**-16.1576**	**-4.486**	**-5.532**	**-10.064**	**-5.063**	-1.315	0.306	**-12.004**	**-17.787**

Table 3: Optimization errors of the CEC2014 benchmark functions (F7-F12) with D=30 & D=50.

Algorithm	function	F7		F8		F9		F10		F11		F12	
	D	30	50	30	50	30	50	30	50	30	50	30	50
LIGSA	Mean	1.25E+00	**3.56E-01**	**4.56E+01**	**1.73E+02**	**5.61E+01**	**2.83E+02**	**1.80E+03**	6.70E+03	5.75E+03	**6.55E+03**	**7.63E-03**	**6.07E-03**
	Best	3.07E-07	1.98E-01	**4.05E+01**	**1.62E+02**	**3.01E+01**	2.55E+02	**1.48E+03**	6.57E+03	5.54E+03	**5.96E+03**	**4.63E-03**	**2.57E-03**
	Dev	2.17E+00	**1.12E-01**	**3.18E+00**	9.18E+00	2.76E+01	2.15E+01	2.57E+02	**1.04E+02**	**2.02E+02**	5.96E+02	**3.00E-03**	**2.59E-03**
	t-test	–	–	–	–	–	–	–	–	–	–	–	–

GSA	Mean	1.00E+01	3.20E+02	1.44E+02	2.68E+02	1.64E+02	3.50E+02	3.73E+03	7.71E+03	4.68E+03	8.12E+03	1.47E+00	2.21E+00
	Best	2.46E+00	2.83E+02	1.37E+02	2.60E+02	1.53E+02	3.11E+02	3.22E+03	7.43E+03	3.94E+03	6.84E+03	1.32E+00	1.84E+00
	Dev	7.64E+00	3.44E+01	6.87E+00	**8.06E+00**	**1.25E+01**	3.53E+01	3.57E+02	1.73E+02	5.56E+02	7.48E+02	1.66E-01	2.62E-01
	t-test	**-2.463**	**-20.793**	**-29.194**	**-17.374**	**-7.968**	**-3.634**	**-9.818**	**-11.135**	4.069	**-3.679**	**-19.700**	**-18.739**
GAGSA	Mean	7.60E+02	1.58E+03	3.70E+02	7.28E+02	3.46E+02	7.86E+02	8.29E+03	1.49E+04	8.73E+03	1.54E+04	3.43E+00	4.08E+00
	Best	6.39E+02	1.56E+03	3.62E+02	7.21E+02	3.30E+02	7.57E+02	7.96E+03	1.40E+04	8.24E+03	1.49E+04	2.71E+00	3.43E+00
	Dev	8.10E+01	1.18E+01	5.26E+00	8.23E+00	1.45E+01	2.49E+01	2.50E+02	6.60E+02	3.37E+02	**2.91E+02**	5.51E-01	3.83E-01
	t-test	**-20.937**	**-300.263**	**-117.899**	**-100.780**	**-20.816**	**-34.216**	**-40.473**	**-27.563**	**-16.953**	**-3.283**	**-13.890**	**-23.752**
PSOGSA	Mean	9.09E+01	2.76E+02	1.14E+02	3.36E+02	2.55E+02	5.32E+02	4.33E+03	8.09E+03	4.56E+03	8.00E+03	1.43E-01	3.08E-01
	Best	3.54E+01	1.40E+02	1.00E+02	2.43E+02	2.18E+02	4.47E+02	3.33E+03	7.32E+03	3.78E+03	7.12E+03	7.45E-02	2.15E-01
	Dev	5.43E+01	7.68E+01	1.40E+01	6.01E+01	2.93E+01	8.78E+01	5.97E+02	8.59E+02	6.61E+02	7.90E+02	6.37E-02	1.15E-01
	t-test	**-3.689**	**-8.035**	**-8.693**	**-6.003**	**-11.070**	**-6.164**	**-8.686**	**-3.573**	3.842	**-29.713**	**-4.742**	**-5.684**
GGSA	Mean	**1.59E-13**	5.69E+00	1.21E+02	2.52E+02	1.28E+02	2.91E+02	3.26E+03	6.72E+03	**3.95E+03**	7.43E+03	8.10E-03	1.37E-02
	Best	**1.14E-13**	2.35E+00	1.14E+02	2.36E+02	1.09E+02	**2.45E+02**	3.01E+03	5.59E+03	**3.55E+03**	6.23E+03	1.01E-03	4.36E-03
	Dev	**6.23E-14**	3.47E+00	5.39E+00	1.49E+01	1.32E+01	3.42E+01	**2.35E+02**	6.55E+02	3.79E+02	7.13E+02	5.19E-03	6.52E-03
	t-test	1.287	**-3.435**	**-26.878**	**-10.052**	**-5.270**	-0.434	**-9.358**	-0.057	9.372	**-2.131**	-0.175	**-2.421**
MGSA	Mean	2.46E-03	9.22E-01	1.14E+02	2.67E+02	1.29E+02	2.98E+02	4.13E+03	**6.59E+03**	4.06E+03	1.19E+04	7.71E-01	1.45E+00
	Best	0.00E+00	**1.28E-02**	1.00E+02	2.27E+02	9.35E+01	2.88E+02	2.84E+03	**5.48E+03**	3.39E+03	1.12E+04	6.21E-02	1.19E-01
	Dev	5.51E-03	5.18E-01	1.40E+01	2.34E+01	2.32E+01	**1.22E+01**	1.04E+03	7.67E+02	5.17E+02	4.62E+02	1.27E+00	1.54E+00
	t-test	1.284	**-2.386**	**-10.679**	**-8.351**	**-4.504**	-1.360	**-4.875**	0.320	6.834	**-15.827**	-1.341	**-2.083**
LPSO	Mean	1.91E+02	8.09E+02	1.98E+02	3.73E+02	1.84E+02	5.09E+02	4.71E+03	9.27E+03	6.53E+03	1.33E+04	2.06E+00	2.14E+00
	Best	1.35E+02	6.52E+02	1.76E+02	3.19E+02	1.33E+02	4.68E+02	4.02E+03	8.01E+03	5.52E+03	1.03E+04	1.12E+00	1.10E+00
	Dev	4.60E+01	1.45E+02	1.63E+01	3.74E+01	5.27E+01	4.62E+01	5.05E+02	7.88E+02	1.09E+03	1.94E+03	8.10E-01	6.37E-01
	t-test	**-9.218**	**-12.4644**	**-20.461**	**-11.6315**	**-4.803**	**-9.926**	**-11.464**	**-7.212**	-1.561	**-7.405**	**-5.677**	**-7.471**

Table 4: Optimization errors of the CEC2014 benchmark functions (F13-F18) with D=30 & D=50.

Algorithm	function	F13		F14		F15		F16		F17		F18	
	D	30	50	30	50	30	50	30	50	30	50	30	50
LIGSA	Mean	**2.34E-01**	4.73E-01	**2.33E-01**	**2.67E-01**	1.34E+01	**3.57E+01**	**1.24E+01**	**2.23E+01**	**1.68E+03**	**1.34E+05**	**6.54E+01**	**1.91E+03**
	Best	**1.75E-01**	4.16E-01	**1.86E-01**	2.49E-01	9.74E+00	**3.40E+01**	**1.09E+01**	**2.17E+01**	**1.47E+03**	**2.77E+04**	**5.85E+01**	**4.45E+02**
	Dev	**3.53E-02**	4.69E-02	3.60E-02	**1.45E-02**	2.19E+00	**1.43E+00**	9.54E-01	4.54E-01	**1.93E+02**	**1.03E+05**	**4.97E+00**	**9.39E+02**
	t-test	–	–	–	–	–	–	–	–	–	–	–	–
GSA	Mean	3.66E-01	3.65E+00	1.58E+00	6.70E+01	6.05E+01	1.98E+04	1.35E+01	2.26E+01	4.96E+06	4.14E+07	6.21E+02	1.04E+07
	Best	3.03E-01	3.48E+00	2.18E-01	5.71E+01	3.30E+01	1.46E+04	1.29E+01	2.23E+01	3.79E+06	7.78E+06	2.59E+02	3.85E+03
	Dev	4.08E-02	1.17E-01	3.00E+00	1.02E+01	2.05E+01	4.20E+03	3.87E-01	3.29E-01	1.19E+06	2.41E+07	5.12E+02	2.33E+07
	t-test	**-5.466**	**-56.369**	-1.000	**-14.632**	**-5.101**	**-10.544**	**-2.284**	-1.104	**-9.356**	**-3.834**	**-2.424**	-1.000
GAGSA	Mean	9.16E+00	8.91E+00	3.25E+02	4.04E+02	4.45E+05	7.65E+06	1.39E+01	2.32E+01	1.77E+08	7.93E+08	6.50E+09	2.22E+10
	Best	8.76E+00	8.58E+00	2.97E+02	3.80E+02	3.94E+05	6.35E+06	1.37E+01	2.31E+01	7.08E+07	6.14E+08	4.91E+09	2.00E+10
	Dev	3.60E-01	1.84E-01	1.66E+01	1.54E+01	3.52E+04	8.19E+05	1.90E-01	**7.31E-02**	8.51E+07	1.69E+08	1.49E+09	3.31E+09

	t-test	**-55.230**	**-99.101**	**-43.627**	**-58.622**	**-28.321**	**-20.897**	**-3.441**	**-4.010**	**-4.653**	**-10.470**	**-9.781**	**-14.988**
PSOGSA	Mean	2.37E+00	3.82E+00	6.34E+01	7.59E+01	1.14E+05	2.53E+06	1.31E+01	2.24E+01	6.40E+06	2.58E+07	8.80E+03	1.81E+09
	Best	6.42E-01	2.16E+00	3.19E+00	3.73E+01	5.27E+01	1.15E+04	1.25E+01	2.22E+01	9.77E+04	4.86E+06	4.92E+02	4.42E+03
	Dev	1.38E+00	1.03E+00	6.59E+01	3.26E+01	2.07E+05	4.41E+06	4.91E-01	1.42E-01	1.00E+07	4.23E+07	1.00E+04	2.93E+09
	t-test	**-3.455**	**-7.263**	**-2.145**	**-5.187**	-1.235	-1.281	-1.368	-0.102	-1.429	-1.357	-1.948	-1.380
GGSA	Mean	2.73E-01	**4.48E-01**	2.46E-01	2.76E-01	1.50E+01	9.40E+02	1.33E+01	2.27E+01	1.94E+06	4.44E+06	3.96E+02	2.75E+03
	Best	2.36E-01	**4.02E-01**	2.27E-01	2.55E-01	1.05E+01	5.71E+02	1.29E+01	2.22E+01	1.52E+06	2.90E+06	1.59E+02	9.98E+02
	Dev	2.96E-02	**4.41E-02**	**1.77E-02**	2.11E-02	5.57E+00	2.37E+02	2.21E-01	4.16E-01	4.83E+05	1.07E+06	1.89E+02	1.48E+03
	t-test	-1.888	0.870	-0.732	-0.758	-0.597	**-8.549**	-2.019	-1.383	**-8.960**	**-8.949**	**-3.909**	-0.799
MGSA	Mean	4.22E-01	5.35E-01	2.54E-01	2.18E+00	**7.18E+00**	1.13E+02	1.25E+01	2.28E+01	1.11E+06	2.05E+06	1.63E+03	1.95E+03
	Best	3.13E-01	4.00E-01	2.24E-01	**2.37E-01**	**4.47E+00**	7.80E+01	1.23E+01	2.20E+01	4.47E+05	1.55E+06	3.45E+02	1.03E+03
	Dev	9.66E-02	1.45E-01	4.09E-02	4.19E+00	**1.92E+00**	2.54E+01	**1.90E-01**	5.09E-01	8.88E+05	4.89E+05	1.43E+03	1.82E+03
	t-test	**-4.079**	-0.905	-0.870	-1.0219	4.753	**-6.829**	**-2.284**	-1.505	**-2.794**	**-8.576**	**-2.444**	-0.037
LPSO	Mean	4.66E+00	5.95E+00	7.84E+01	2.23E+02	2.93E+04	1.33E+06	1.33E+01	2.26E+01	1.25E+07	4.05E+07	1.26E+09	5.39E+09
	Best	4.26E+00	5.54E+00	3.42E+01	1.79E+02	4.21E+03	6.00E+04	1.30E+01	2.21E+01	2.44E+05	1.99E+07	5.03E+08	3.01E+09
	Dev	3.03E-01	4.13E-01	2.96E+01	4.37E+01	3.09E+04	1.94E+06	2.67E-01	4.74E-01	2.35E+07	2.28E+07	5.23E+08	1.84E+09
	t-test	**-32.463**	**-29.434**	**-5.914**	**-11.394**	**-2.126**	-1.542	**-2.093**	-1.029	-1.187	**-3.960**	**-5.395**	**-6.565**

Table 5: Optimization errors of the CEC2014 benchmark functions (F19-F24) with D=30 & D=50.

Algorithm	function	F19		F20		F21		F22		F23		F24	
	D	30	50	30	50	30	50	30	50	30	50	30	50
LIGSA	Mean	**6.94E+00**	**2.15E+01**	**4.30E+01**	**1.21E+03**	**8.93E+02**	**1.44E+04**	**1.40E+02**	**1.32E+03**	**2.00E+02**	**2.00E+02**	**2.00E+02**	**2.00E+02**
	Best	**5.76E+00**	**1.56E+01**	**3.92E+01**	**7.54E+02**	**5.54E+02**	**5.84E+03**	**4.67E+01**	**1.09E+03**	**2.00E+02**	**2.00E+02**	**2.00E+02**	**2.00E+02**
	Dev	**1.19E+00**	**1.07E+01**	**3.05E+00**	**5.49E+02**	**1.93E+02**	**5.76E+03**	**1.15E+02**	**1.65E+02**	**7.99E-10**	**8.39E-10**	**8.70E-07**	**8.40E-07**
	t-test	–	–	–	–	–	–	–	–	–	–	–	–
GSA	Mean	6.78E+01	2.14E+02	1.58E+05	8.35E+04	1.63E+06	4.65E+06	1.06E+03	2.16E+03	2.25E+02	**2.00E+02**	**2.00E+02**	2.41E+02
	Best	3.01E+01	1.38E+02	1.18E+05	6.31E+04	4.88E+05	3.21E+06	4.42E+02	1.80E+03	**2.00E+02**	**2.00E+02**	**2.00E+02**	2.31E+02
	Dev	3.17E+01	7.29E+01	3.01E+04	1.90E+04	7.64E+05	1.88E+06	5.24E+02	2.39E+02	5.67E+01	1.28E-08	6.36E-01	6.26E+00
	t-test	**-4.292**	**-5.837**	**-11.749**	**-9.687**	**-4.757**	**-5.501**	**-3.821**	**-6.466**	-1.000	**-30.382**	-1.531	**-14.467**
GAGSA	Mean	6.19E+02	4.21E+03	7.76E+06	1.82E+05	1.36E+08	2.69E+08	5.62E+03	5.27E+05	5.93E+02	1.12E+03	2.39E+02	3.33E+02
	Best	5.26E+02	3.12E+03	9.36E+05	1.70E+05	5.35E+07	1.05E+08	2.76E+03	1.95E+05	3.15E+02	1.07E+03	2.28E+02	3.20E+02
	Dev	5.41E+01	7.37E+02	1.34E+07	8.51E+03	8.34E+07	1.10E+08	4.27E+03	4.14E+05	1.79E+02	4.86E+01	7.33E+00	1.27E+01
	t-test	**-25.259**	**-12.696**	-1.295	**-47.409**	**-3.635**	**-5.475**	**-2.870**	**-2.841**	**-4.917**	**-42.126**	**-11.890**	**-23.480**
PSOGSA	Mean	1.37E+02	2.26E+02	4.36E+04	1.04E+05	1.96E+06	2.80E+06	1.07E+03	1.93E+03	3.89E+02	7.29E+02	2.79E+02	3.95E+02
	Best	9.69E+01	1.46E+02	8.08E+03	8.15E+04	5.68E+05	4.17E+05	6.78E+02	1.29E+03	3.42E+02	4.36E+02	2.51E+02	3.67E+02
	Dev	5.39E+01	7.64E+01	5.03E+04	1.82E+04	2.04E+06	2.34E+06	3.13E+02	7.04E+02	3.58E+01	2.77E+02	2.86E+01	2.20E+01
	t-test	**-5.382**	**-5.943**	-1.934	**-12.571**	**-2.146**	**-2.666**	**-6.218**	-1.880	**-11.804**	**-4.263**	**-6.186**	**-19.863**
GGSA	Mean	3.97E+01	5.34E+01	3.84E+04	4.59E+04	2.40E+05	2.92E+06	9.89E+02	2.12E+03	3.21E+02	3.30E+02	**2.00E+02**	2.45E+02
	Best	1.71E+01	3.68E+01	3.29E+04	3.94E+04	1.78E+05	1.59E+06	8.11E+02	1.62E+03	3.19E+02	**2.00E+02**	**2.00E+02**	2.25E+02
	Dev	2.70E+01	1.99E+01	5.50E+03	5.57E+03	5.43E+04	1.02E+06	1.95E+02	5.03E+02	1.04E+00	1.80E+02	8.70E-07	1.29E+01
	t-test	**-2.712**	**-3.157**	**-15.581**	**-17.861**	**-9.841**	**-6.342**	**-8.379**	**-3.376**	**-258.577**	-1.617	-1.602	**-7.845**
MGSA	Mean	2.55E+01	6.26E+01	3.63E+04	4.86E+04	3.72E+05	2.45E+06	7.31E+02	1.70E+03	3.16E+02	3.68E+02	2.45E+02	3.00E+02
	Best	1.25E+01	2.23E+01	1.61E+04	2.36E+04	2.40E+05	7.67E+05	2.94E+02	1.41E+03	3.15E+02	3.57E+02	2.33E+02	2.87E+02
	Dev	2.46E+01	3.60E+01	1.69E+04	2.22E+04	1.16E+05	1.10E+06	3.42E+02	1.82E+02	4.86E-01	1.42E+01	6.76E+00	8.21E+00
	t-test	-1.681	**-2.447**	**-4.795**	**-4.776**	**-7.155**	**-4.980**	**-3.663**	**-3.433**	**-532.918**	**-26.428**	**-14.792**	**-27.346**
LPSO	Mean	1.64E+02	7.68E+02	4.72E+04	1.60E+05	2.98E+05	1.13E+07	1.05E+03	2.09E+03	5.31E+02	7.90E+02	2.94E+02	4.69E+02
	Best	8.78E+01	4.53E+02	1.37E+04	7.22E+04	2.21E+04	4.74E+06	5.73E+02	1.72E+03	3.36E+02	6.62E+02	2.66E+02	4.36E+02
	Dev	6.26E+01	2.89E+02	3.27E+04	6.53E+04	1.87E+05	7.62E+06	3.53E+02	3.37E+02	1.48E+02	1.86E+02	2.52E+01	1.97E+01
	t-test	**-5.606**	**-5.766**	**-3.218**	**-5.422**	**-3.558**	**-3.325**	**-5.472**	**-4.573**	**-4.986**	**-7.101**	**-8.313**	**-30.470**

Table 6: Optimization errors of the CEC2014 benchmark functions (F25-F30) with D=30 & D=50.

Algorithm	function	F25		F26		F27		F28		F29		F30	
	D	30	50	30	50	30	50	30	50	30	50	30	50
LIGSA	Mean	**2.00E+02**	2.27E+02	1.40E+02	**1.62E+02**	3.96E+02	8.33E+02	**2.00E+02**	**2.00E+02**	9.38E+02	**2.00E+02**	**2.00E+02**	**2.00E+02**
	Best	**2.00E+02**	2.20E+02	**1.00E+02**	**1.00E+02**	3.75E+02	6.90E+02	**2.00E+02**	**2.00E+02**	7.50E+02	**2.00E+02**	**2.00E+02**	**2.00E+02**
	Dev	**3.39E-11**	6.43E+00	5.50E+01	5.65E+01	1.66E+01	1.64E+02	**2.20E-10**	**5.70E-09**	1.57E+02	**4.09E-03**	**8.36E-05**	**5.60E-04**
	t-test	–	–	–	–	–	–	–	–	–	–	–	–
GSA	Mean	2.01E+02	2.03E+02	2.00E+02	2.00E+02	1.61E+03	2.96E+03	1.77E+03	5.87E+03	2.00E+02	**2.00E+02**	3.15E+05	4.78E+06
	Best	**2.00E+02**	2.00E+02	2.00E+02	2.00E+02	1.47E+03	2.53E+03	4.39E+02	5.13E+03	2.00E+02	**2.00E+02**	**2.00E+02**	2.00E+02
	Dev	2.30E+00	4.62E+00	1.54E-02	9.76E-02	2.08E+02	4.13E+02	9.23E+02	4.36E+02	2.62E-02	2.14E-02	1.83E+05	5.50E+06
	t-test	-1.284	**-6.616**	**-2.431**	-1.506	**-12.988**	**-10.700**	**-3.792**	**-29.088**	10.523	**-35.392**	**-3.846**	-1.942
GAGSA	Mean	2.09E+02	**2.00E+02**	2.00E+02	2.00E+02	**2.00E+02**	**2.00E+02**	1.77E+03	1.40E+03	**2.00E+02**	7.23E+06	1.93E+03	9.75E+03
	Best	2.07E+02	**2.00E+02**	2.00E+02	2.00E+02	**2.00E+02**	**2.00E+02**	4.39E+02	1.23E+03	**2.00E+02**	4.97E+03	1.16E+03	9.17E+03
	Dev	2.59E+00	**6.70E-11**	0.00E+00	**4.55E-13**	**2.16E-10**	**3.43E-10**	9.23E+02	1.06E+02	**2.52E-03**	1.62E+07	6.87E+02	5.74E+02
	t-test	**-8.053**	1.676	**-2.429**	-1.498	26.391	8.640	**-46.101**	**-25.385**	10.525	-1.001	**-5.617**	**-37.210**
PSOGSA	Mean	2.29E+02	2.71E+02	1.65E+02	1.55E+02	1.06E+03	1.80E+03	1.90E+03	4.60E+03	6.01E+06	7.03E+07	6.30E+04	5.03E+05
	Best	2.09E+02	2.28E+02	1.01E+02	1.08E+02	1.01E+03	1.70E+03	1.33E+03	3.37E+03	1.06E+05	1.63E+07	7.16E+03	9.73E+04
	Dev	3.10E+01	6.24E+01	5.57E+01	6.21E+01	5.56E+01	8.73E+01	3.86E+02	1.29E+03	7.65E+06	4.86E+07	1.08E+05	3.84E+05
	t-test	**-2.058**	**-2.395**	-0.697	0.192	**-25.750**	**-11.696**	**-9.823**	**-7.619**	-1.756	**-3.234**	-1.296	**-2.927**
GGSA	Mean	**2.00E+02**	2.00E+02	2.00E+02	2.00E+02	1.68E+03	3.22E+03	2.93E+03	5.41E+03	1.30E+03	8.25E+07	7.32E+04	2.62E+05
	Best	**2.00E+02**	2.00E+02	2.00E+02	2.00E+02	8.77E+02	3.02E+03	2.48E+03	3.69E+03	2.00E+02	2.01E+02	2.96E+04	2.00E+02
	Dev	3.71E-10	2.23E-01	**1.19E-02**	6.44E-02	4.84E+02	2.55E+02	5.32E+02	1.25E+03	1.21E+03	1.85E+08	3.55E+04	1.56E+05
	t-test	**-9.211**	1.625	**-2.431**	-1.505	**-5.918**	**-17.565**	**-11.484**	**-9.310**	-0.675	-1.000	**-4.593**	**-3.758**
MGSA	Mean	2.20E+02	2.51E+02	**1.20E+02**	2.01E+02	9.88E+02	1.65E+03	3.61E+03	7.10E+03	3.81E+06	1.23E+05	1.52E+04	1.75E+05
	Best	2.15E+02	2.44E+02	**1.00E+02**	2.01E+02	4.11E+02	1.57E+03	2.83E+03	6.03E+03	2.47E+03	3.64E+03	5.70E+03	8.81E+04
	Dev	2.91E+00	9.09E+00	4.47E+01	2.50E-01	3.38E+02	8.40E+01	7.49E+02	9.90E+02	5.22E+06	2.35E+05	8.08E+03	1.42E+05
	t-test	**-15.438**	**-10.376**	0.631	-1.536	**-3.915**	**-9.956**	**-10.174**	**-15.598**	-1.635	-1.169	**-4.159**	**-2.758**
LPSO	Mean	2.35E+02	2.92E+02	1.07E+02	1.42E+02	1.15E+03	2.08E+03	1.55E+03	3.23E+03	2.38E+07	1.94E+08	1.66E+05	2.44E+06
	Best	2.18E+02	2.65E+02	1.04E+02	1.09E+02	5.50E+02	1.83E+03	1.44E+03	1.95E+03	1.44E+07	1.24E+08	7.06E+04	3.41E+05
	Dev	1.36E+01	3.01E+01	2.48E+00	6.59E+01	3.36E+02	1.55E+02	7.02E+01	1.12E+03	1.01E+07	5.04E+07	1.70E+05	2.86E+06
	t-test	**-5.686**	**-6.497**	1.359	0.517	**-5.014**	**-12.327**	**-43.166**	**-6.042**	**-5.296**	**-8.621**	**-2.178**	-1.903

Table 7: Summary of *t*-test at 5% significant level.

	LIGSA VS.	GSA	GAGSA	PSOGSA	GGSA	MGSA	LPSO
D=30	better	23	27	19	19	20	26
	same	3	1	9	9	8	4
	worse	4	2	2	2	2	0
D=50	better	25	26	21	19	21	25
	same	4	3	8	10	8	4
	worse	1	1	1	1	1	1

As to the *hybrid function 1* (F17-F22), the superiority of LIGSA is more notable. As displayed in Tables 4-5, on both low and high dimensions, LIGSA yield the best results on all the functions in respect of all the three performance metrics. Moreover, the t-test value revealed that the superiority of LIGSA is significant in most cases.

For the last 8 *composition benchmark functions*, F23-F30, due to their complexity in finding the global optima, the preponderance of LIGSA is not as obvious as it done on the other functions. Even though, LIGSA produced best results on 5 out of the 8 functions when D=30 and 6 out of the 8 functions when D=50. The statistically results also confirmed the preponderance of LIGSA. In addition, it is notable that GAGSA performed certain advantages on several *composition benchmark functions*, such as F25, F27, and F29. This may be due to its fitness-free mutation operator can preserve the diversity of population effectively though it improves the computational complexity.

Besides, the summary of t-test at 5% significant level shown in Table 7 reveals the superiority of LIGSA more intuitively. For example, LIGSA outperforms MGSA in optimization of 93.3% (28/30) of the 30-D problems and 96.7% (29/30) of the 50-D problems, where the proportions of significant superiority are 71.4% (20/28) and 72.4 (21/29), respectively.

4.2. Convergence comparison

To test the convergence speed of the proposed LIGSA, the average numbers of fitness evaluation (FES_{mean}) and the execution time (in second) that the algorithms need to obtain acceptable solutions should be reported. In this paper, the trial is considered to be acceptable if and only if the *error* satisfies $error \leq \varepsilon_{\text{opt}}$ where ε_{opt} was set to 0.01. Moreover, an effective algorithm should perform good search reliability, i.e., obtain acceptable successful rate "suc%". The "suc%" stands for the percentage of the successful runs that acceptable solutions are found [6]. Due to the CEC2014 benchmark functions are quite complexity, only functions F7 and F12 can be solved by several of the 7 algorithms. The corresponding results are reported in Table 8.

Table 8: Convergence speed and reliability comparisons on F7 and F12.

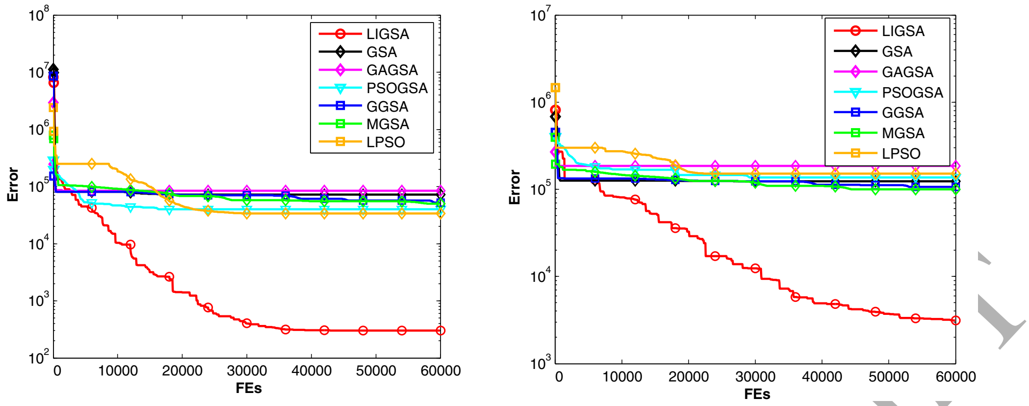
	Algorithm	LIGSA		GSA		GAGSA		PSOGSA		GGSA		MGSA		LPSO	
	D	30	50	30	50	30	50	30	50	30	50	30	50	30	50
F7	FES_{mean}	**57600**	N/A	N/A	N/A	N/A	N/A	N/A	N/A	83220	N/A	61440	N/A	N/A	N/A
	time	**6.962**	N/A	N/A	N/A	N/A	N/A	N/A	N/A	15.930	N/A	12.051	N/A	N/A	N/A
	suc%	40	N/A	N/A	N/A	N/A	N/A	N/A	N/A	**100**	N/A	80	N/A	N/A	N/A
	rank	1	1	4	1	4	1	4	1	3	1	2	1	4	1
F12	FES_{mean}	**31440**	**36240**	N/A	N/A	N/A	N/A	N/A	N/A	71400	89820	N/A	N/A	N/A	N/A
	time	**6.046**	**7.929**	N/A	N/A	N/A	N/A	N/A	N/A	14.408	20.446	N/A	N/A	N/A	N/A
	suc%	**80**	**100**	N/A	N/A	N/A	N/A	N/A	N/A	40	**100**	N/A	N/A	N/A	N/A
	rank	1	1	3	3	3	3	3	3	2	2	3	3	3	3

In Table 8, the ranks are evaluated based on the ascending order of FES_{mean}. According to Table 8, for F7, when D=30, LIGSA, GGSA and MGSA can obtain acceptable solutions. Among the three algorithms, GGSA performs the best reliability while LIGSA shows the fast convergence speed as its takes the smallest FES_{mean} and shortest consuming time. With respect to high dimension (D=50), none of the algorithms can yield acceptable solutions. For F12, on the 30-D and 50-D problems, LIGSA and GGSA are the only two effective algorithms. The metrics FES_{mean}, time, and suc% verify the better convergence performance of LIGSA on the function.

For the other 28 CEC2014 functions, although none of the tested algorithms can obtain acceptable solutions, the superior convergence behavior of LIGSA can also be illustrated by their convergence curve. Due to the large number of functions in the CEC2014 problem set, we selected one function from each category to illustrate the convergence of the compared algorithms. The selected functions are F3, F16, F21 and F28. Their convergence curves are presented in Figs. 6-9, respectively. As shown in Fig. 7 and Fig. 9, LIGSA achieved the fastest convergence speed on F16 and F28. For functions F3 and F21 as shown in Fig. 6 and Fig. 8 respectively, although the fall-off rate LIGSA is not the biggest in the early iterations, it keeps searching for a better solution for longest time and produced the best optimization results.

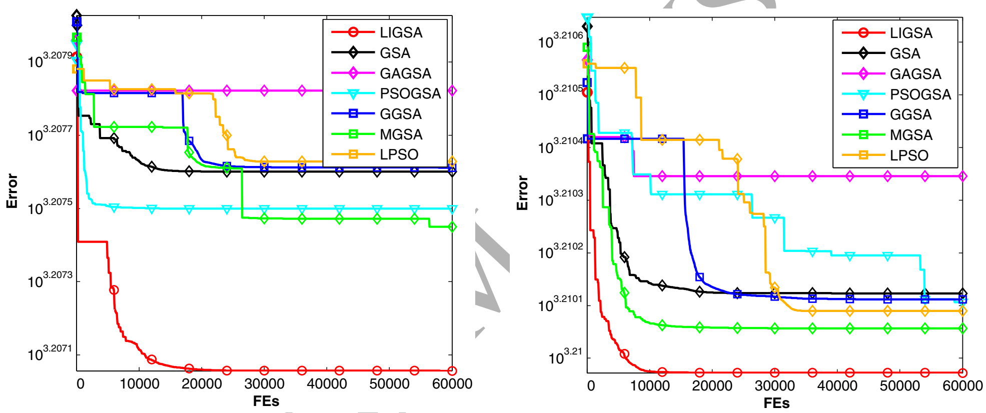
4.3. Discussion

Achieving a fine balance of exploration and exploitation is challenging for all meta-heuristic algorithms [39]. In basic GSA, a neighborhood topology denoted by K_{best} is employed to get the balance. However, the diminishing number



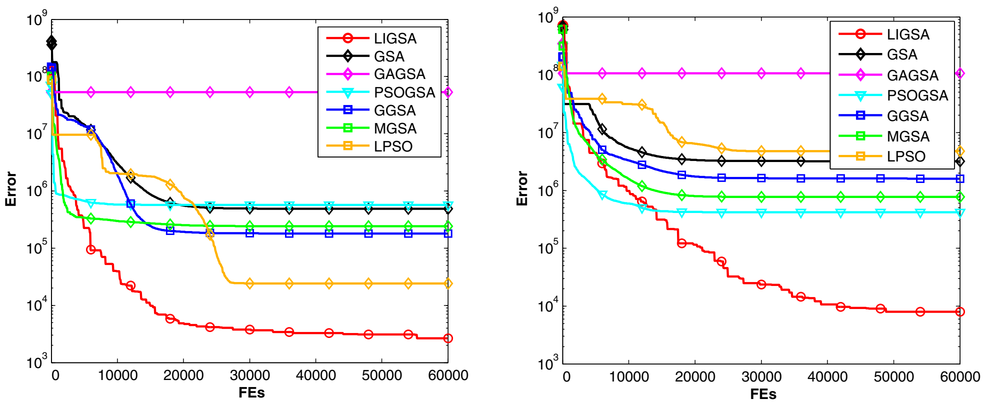
(a) D=30 (b) D=50

Figure 6: Convergence performance comparison for minimizing of F3.



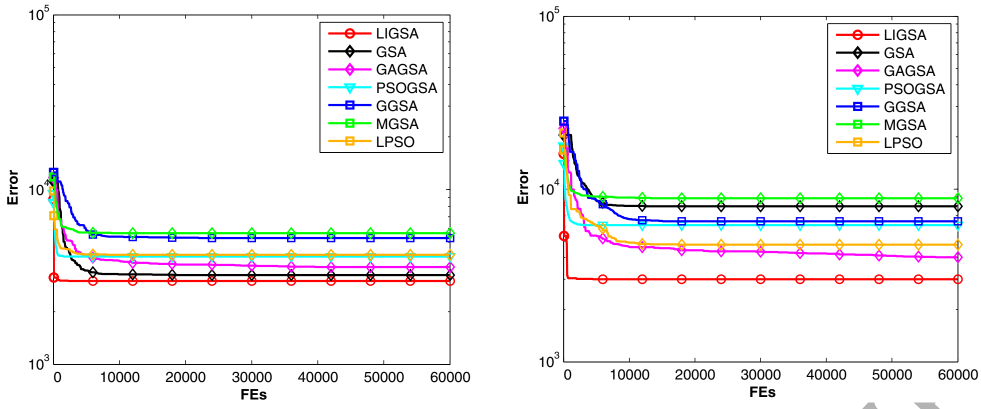
(a) D=30 (b) D=50

Figure 7: Convergence performance comparison for minimizing of F16.



(a) D=30 (b) D=50

Figure 8: Convergence performance comparison for minimizing of F21.



(a) D=30 (b) D=50

Figure 9: Convergence performance comparison for minimizing of F28.

of K_{best} leads to premature convergence over quickly. Moreover, K_{best} is a global neighborhood topology in essence. This type of topology lacks of local search ability and is time consuming [21]. More importantly, the K_{best} model is a kind of fully-informed method. All agents choose the same neighbors stored in K_{best}, which overlooking the impact of environmental heterogeneity on individual behavior. As a result, GSA is easily to suffer from premature convergence. As illustrated in Figs. 6-9 and Tables 2-7 of Section 4.2, the convergence curve and accuracy confirmed the insufficient local search problem of GSA.

By analysis the convergence curves of the LIGSA on D=30 and D=50 problems in Figs. 6-9 and Table 8, one may conclude that the LIGSA performed faster convergence speed (with smallest FES_{mean}) on the tested functions. This suggests that the LIGSA has outstanding global search ability. It is due to the introduction of the social thinking improves the global search ability of LIGSA and accelerate its convergence speed.

In addition, once LIGSA successfully obtained acceptable solution on a function, it achieved the best convergence accuracy with lowest runtime consuming as illustrated in Table 8. This property is due to the proposed locally informed learning strategy completely changed the global neighborhood topology structure in fully-informed learning strategy and preserved the diverse search directions of GSA. In other words, the individual heterogeneity is taken into consideration in LIGSA. Therefore, LIGSA improves the local search ability and reduces the computational complexity.

Although there are many important discoveries revealed by these studies, there are also limitations. That is, all these tested algorithms are hard to achieve the precision requirement in preset evaluation times. Although LIGSA produced a better optimization results, the search ability still required great improvement. One possible reason is that GSA does not have invariance in shift, rotation, and scale for gravitational force between agents is highly influenced by the distance between them.

In future work, we intend to analysis the invariance of other physics-inspired meta-heuristic optimization algorithms, such as electromagnetic field optimization (EFO) [1], in shift, rotation, and scale to improve the search ability of LIGSA and solve more shifted and rotated functions.

5. Conclusion

In this paper, taking into account the heterogeneity of individuals behaviors we proposed a locally informed GSA (LIGSA) to enhance the search ability of the original GSA. The LIGSA was characterized by designing a locally informed learning strategy. In this learning strategy, a locally informed k-neighborhood was developed for constructing unique local neighborhood for every agent, thus promoting LIGSA fully explore the search space with low computational complexity as well as effectively prevent premature convergence; the social thinking of population was employed to guide all the agents and accelerate the convergence speed; the time-varying acceleration coefficients scheme was proposed to balance the two components and promote robustness of GSA. Therefore, LIGSA could significantly improve the performance of GSA.

To investigate the validity of LIGSA, all the 30 CEC2014 benchmark functions with both low and high dimensions were tested in this paper. The graphical and statistical results were compared with the original GSA, four variants of GSA, and LPSO. The compared experimental results demonstrated the significant superiority of LIGSA in most cases. Moreover, although the search availability of LIGSA requires further promote, LIGSA generally showed more rapidly convergence ability, lower computational complexity, and higher convergence accuracy.

Acknowledgements

This work was supported by Chinese Natural Science Foundation Projects (41471353, 41271349), Fundamental Research Funds for the Central Universities (14CX02039A, 15CX06001A), and the China Scholarship Council. The authors are very grateful to all anonymous reviewers whose invaluable comments and suggestions substantially helped improve the quality of the paper.

References

[1] Abedinpourshotorban, H., Shamsuddin, S.M., Beheshti, Z., Jawawi, D.N., 2015. Electromagnetic field optimization: A physics-inspired metaheuristic optimization algorithm. Swarm and Evolutionary Computation doi:10.1016/j.swevo.2015.07.002.

[2] Beheshti, Z., Shamsuddin, S.M.H., 2014. CAPSO: centripetal accelerated particle swarm optimization. Information Sciences 258, 54–79. doi:10.1016/j.ins.2013.08.015.

[3] Beheshti, Z., Shamsuddin, S.M.H., Hasan, S., 2013. MPSO: median-oriented particle swarm optimization. Applied Mathematics and Computation 219, 5817–5836. doi:10.1016/j.amc.2012.12.013.

[4] Brooks, S.P., Morgan, B.J., 1995. Optimization using simulated annealing. The Statistician , 241–257doi:10.2307/2348448.

[5] Chelouah, R., Siarry, P., 2003. Genetic and nelder–mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions. European Journal of Operational Research 148, 335–348. doi:10.1016/S0377-2217(02)00401-0.

[6] Chen, W.N., Zhang, J., Lin, Y., Chen, N., Zhan, Z.H., Chung, H.H., Li, Y., Shi, Y.H., 2013. Particle swarm optimization with an aging leader and challengers. Evolutionary Computation, IEEE Transactions on 17, 241–258. doi:10.1109/TEVC.2011.2173577.

[7] Civicioglu, P., 2012. Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm. Computers & Geosciences 46, 229–247. doi:10.1016/j.cageo.2011.12.011.

[8] Civicioglu, P., 2013a. Artificial cooperative search algorithm for numerical optimization problems. Information Sciences 229, 58–76. doi:10.1016/j.ins.2012.11.013.

[9] Civicioglu, P., 2013b. Backtracking search optimization algorithm for numerical optimization problems. Applied Mathematics and Computation 219, 8121–8144. doi:10.1016/j.amc.2013.02.017.

[10] Doraghinejad, M., Nezamabadi-pour, H., 2014. Black hole: A new operator for gravitational search algorithm. International Journal of Computational Intelligence Systems 7, 809–826.

[11] Dorigo, M., Maniezzo, V., Colorni, A., 1996. Ant system: optimization by a colony of cooperating agents. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 26, 29–41. doi:10.1109/3477.484436.

[12] Eberhart, R.C., Shi, Y., 2001. Particle swarm optimization: developments, applications and resources, in: Evolutionary Computation, 2001. Proceedings of the 2001 Congress on, IEEE. pp. 81–86. doi:10.1109/CEC.2001.934374.

[13] Gu, B., Pan, F., 2013. Modified gravitational search algorithm with particle memory ability and its application. International Journal of Innovative Computing, Information and Control 9, 4531–4544.

[14] Holland, J.H., 1975. Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. U Michigan Press.

[15] Hu, X., Shi, Y., Eberhart, R., 2004. Recent advances in particle swarm, in: IEEE congress on evolutionary computation, Portland. pp. 90–97. doi:10.1109/CEC.2004.1330842.

[16] Jiang, S., Ji, Z., Shen, Y., 2014a. A novel hybrid particle swarm optimization and gravitational search algorithm for solving economic emission load dispatch problems with various practical constraints. International Journal of Electrical Power & Energy Systems 55, 628 – 644. doi:10.1016/j.ijepes.2013.10.006.

[17] Jiang, S., Wang, Y., Ji, Z., 2014b. Convergence analysis and performance of an improved gravitational search algorithm. Applied Soft Computing 24, 363–384. doi:10.1016/j.asoc.2014.07.016.

[18] Kaveh, A., Laknejadi, K., 2011. A novel hybrid charge system search and particle swarm optimization method for multi-objective optimization. Expert Systems with Applications 38, 15475–15488. doi:10.1016/j.eswa.2011.06.012.

[19] Kennedy, J., 2006. Handbook of nature-inspired and innovative computing. Swarm intelligence. Springer , 187–219.

[20] Kennedy, J., Eberhart, R., et al., 1995. Particle swarm optimization, in: Proceedings of IEEE international conference on neural networks, Perth, Australia. pp. 1942–1948.

[21] Kennedy, J., Kennedy, J.F., Eberhart, R.C., 2001. Swarm intelligence. Morgan Kaufmann.

[22] Li, X., Yin, M., Ma, Z., 2011. Hybrid differential evolution and gravitation search algorithm for unconstrained optimization. Int. J. Phys. Sci 6, 5961–5981. doi:10.5897/IJPS11.029.

[23] Liang, J., Qu, B., Suganthan, P., 2013. Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore .

[24] Mendes, R., Kennedy, J., Neves, J., 2004. The fully informed particle swarm: simpler, maybe better. Evolutionary Computation, IEEE Transactions on 8, 204–210. doi:10.1109/TEVC.2004.826074.

[25] Mirjalili, S., Hashim, S.Z.M., 2010. A new hybrid PSOGSA algorithm for function optimization, in: Computer and information application (ICCIA), 2010 international conference on, IEEE. pp. 374–377. doi:10.1109/ICCIA.2010.6141614.

[26] Mirjalili, S., Hashim, S.Z.M., Sardroudi, H.M., 2012. Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm. Applied Mathematics and Computation 218, 11125–11137. doi:10.1016/j.amc.2012.04.069.

[27] Mirjalili, S., Lewis, A., 2014. Adaptive gbest-guided gravitational search algorithm. Neural Computing and Applications 25, 1569–1584. doi:10.1007/s00521-014-1640-y.

[28] Nasir, M., Das, S., Maity, D., Sengupta, S., Halder, U., Suganthan, P.N., 2012. A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization. Information Sciences 209, 16–36. doi:10.1016/j.ins.2012.04.028.

[29] Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S., 2009. GSA: a gravitational search algorithm. Information sciences 179, 2232–2248. doi:10.1016/j.ins.2009.03.004.

[30] Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S., 2010. BGSA: binary gravitational search algorithm. Natural Computing 9, 727–745. doi:10.1007/s11047-009-9175-3.

[31] Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S., 2011. Filter modeling using gravitational search algorithm. Engineering Applications of Artificial Intelligence 24, 117–122. doi:10.1016/j.engappai.2010.05.007.

[32] Sarafrazi, S., Nezamabadi-Pour, H., Saryazdi, S., 2011. Disruption: a new operator in gravitational search algorithm. Scientia Iranica 18, 539–548. doi:10.1016/j.scient.2011.04.003.

[33] Shaw, B., Mukherjee, V., Ghoshal, S., 2012. A novel opposition-based gravitational search algorithm for combined economic and emission dispatch problems of power systems. International Journal of Electrical Power & Energy Systems 35, 21–33. doi:10.1016/j.ijepes.2011.08.012.

[34] Simon, D., 2008. Biogeography-based optimization. Evolutionary Computation, IEEE Transactions on 12, 702–713. doi:10.1109/TEVC.2008.919004.

[35] Sun, G., Zhang, A., 2013. A hybrid genetic algorithm and gravitational search algorithm for image segmentation using multilevel thresholding, in: Pattern Recognition and Image Analysis. Springer, pp. 707–714. doi:10.1007/978-3-642-38628-2_84.

[36] Tran, D.H., Cheng, M.Y., Cao, M.T., 2015. Hybrid multiple objective artificial bee colony with differential evolution for the time–cost–quality tradeoff problem. Knowledge-Based Systems 74, 176–186. doi:10.1016/j.knosys.2014.11.018.

[37] Tsai, H.C., Tyan, Y.Y., Wu, Y.W., Lin, Y.H., 2013. Gravitational particle swarm. Applied Mathematics and Computation 219, 9106–9117. doi:10.1016/j.amc.2013.03.098.

[38] Wang, H., Sun, H., Li, C., Rahnamayan, S., Pan, J.S., 2013. Diversity enhanced particle swarm optimization with neighborhood search. Information Sciences 223, 119–135. doi:10.1016/j.ins.2012.10.012.

[39] Yang, X.S., Deb, S., He, X., 2013. Eagle strategy with flower algorithm, in: Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on, IEEE. pp. 1213–1217. doi:10.1109/ICACCI.2013.6637350.

[40] Yazdani, S., Nezamabadi-pour, H., Kamyab, S., 2014. A gravitational search algorithm for multimodal optimization. Swarm and Evolutionary Computation 14, 1–14. doi:10.1016/j.swevo.2013.08.001.

[41] Zhang, W., Niu, P., Li, G., Li, P., 2013. Forecasting of turbine heat rate with online least squares support vector machine based on gravitational search algorithm. Knowledge-Based Systems 39, 34–44. doi:10.1016/j.knosys.2012.10.004.